REMARKS

Claims 1, 2, 4 – 9, and 11 – 14 are pending in this application.

In an Office Action mailed 04 March 2008, the Examiner rejected claims 1, 2, 4 – 9, and 11 – 14 under 35 USC §102(e) as being anticipated by Harrison (US Patent No. 6,901,582, hereinafter "the Harrison Patent"). The Examiner noted with respect to the independent claims:

> Harrison teaches:
>
> Software application selection means, , responsive to user input, for identifying a software application, equipped with a Reporting Application Interface, that is executing on said processor (see at least Fig. 1a-1e; also Fig. 3);
>
> Function Identification means for automatically determining which function is presently executing in said identified software application (see at least Fig. 1a-1e; also see at least col. 6, lines 21-25 "The visual map 120 comprises a snapshot high-level overview of on-screen graphics representing major application components (i.e. functions) and groups of application components of the application being monitored");
>
> Polling means for bypassing said Reporting Application Interface, and periodically polling said presently executing function in said identified software application to retrieve statistics data indicative of operation of said presently executing function in said identified software application (see at least col. 10, lines 65-67 "According to an embodiment, the monitoring system 220 periodically issues an SQL statement requesting data from DBMS 300 representing performance statistics of the DBMS 300, stored in the database files 335"; also see col. 19, lines 1-7 "the monitoring system 220 may also employ software modules designed to extract some or all of the performance statistics without accessing or calling the API. By avoiding a call to the API, the software modules extracts the performance statistics without adding significant program loads to the application, thereby advantageously allowing for more frequent polling, more precise polling, or both"); and
>
> Data repository means for storing data indicative of an identity of said presently executing function and said retrieved statistics data in a memory (see at least Fig. 1a-1e and Fig. 3; see also col. 11, line 1 "performance statistics of the DBMS 300, stored in the database files 335").

Applicant has reviewed the cited Harrison Patent and presents the following arguments to properly distinguish Applicant's invention from the teachings of the Harrison Patent.

In particular, the Harrison Patent teaches a method for monitoring the performance of an application using a standard monitoring system graphical user interface (GUI) having an interactive window with on-screen graphics representative of the application architecture and component performance. The GUI comprises straightforward and intuitive icons, panels, and data flows, representing the existence, capacity, or number of processes, memory structures, and the like of the application being monitored. In addition, the monitoring system includes an alert firing mechanism and process for alerting an administrator to potential or actual performance problems in the application.

Thus, the Harrison GUI is the proprietary Reporting Application Interface to periodically poll the executing software application to perform a time-wise snapshot of the executing software application thereby to limit the amount of data generated. As described in column 4, lines 52 – 59, of the Harrison Patent:

> FIGS. 1A-1E illustrate various embodiments of monitoring system GUIs 100, according to aspects of the invention. The GUIs 100 govern the way in which a user and the monitoring systems interact. For example, the user may wish to input commands to the monitoring systems, while the monitoring systems provide both feedback of the user commands and provide a summary of the current and past performance of the respective applications being monitored.

In the monitoring system of the Harrison Patent, the application performance data is collected, stored in a database DBMS 220, and then retrieved by the monitoring system 220 to create the displays and standard reports for the user. This is described in column 10, line 65 – column 11, line 5 of the Harrison Patent:

> According to an embodiment, the monitoring system 220 periodically issues an SQL statement requesting data from the DBMS 300 representing performance statistics of the DBMS 300, stored in the database files 335. Similar to the foregoing data request, the server process 310 accesses the data either through the buffer cache 320 of the database files 335. The server process 310 then returns the performance statistics to the monitoring system 220.

The only departure from this standard monitoring system operation is noted in column 18, line 62 – column 19, line 7 of the Harrison Patent:

> In addition, a skilled artisan will recognize from the disclosure herein that the monitoring system may advantageous access some or all of the performance statistics through, for example, an open application program interface (API) of the application being monitored. On the other hand, a skilled artisan will recognize from the disclosure herein that the monitoring system 220 may also employ software modules designed to extract some or all of the performance statistics without accessing or calling the API. By avoiding a call to the API, the software modules extract the performance statistics without adding significant

> program loads to the application, thereby advantageously allowing for more frequent polling, more precise polling, or both.

In contrast, Applicant's system for monitoring operation of a software application bypasses the proprietary Reporting Application Interface to periodically poll the executing software application to perform a time-wise snapshot of the executing software application thereby to limit the amount of data generated, yet provide a more comprehensive view of the executing software application than an error reporting system as in the Harrison Patent. Applicant's polling system periodically checks the software application to determine which function is presently executing within the software application. The periodic polling of executing processes significantly reduces the amount of data collected, yet enables the application stakeholder to obtain a granularity of data relating to the executing software application not available with existing software monitoring systems. The function polling system monitors all the functions embodied in the software application rather than selected functions thereby to avoid gaps in the coverage of the monitoring.

Thus, Applicant's system continuously monitors the running software application and identifies which function in this software application is presently executing at every polling increment of time. This process, therefore, produces a **continuum of data**, by creating periodic snapshots of the present operation of the software and its executing functions, so the user can view the history of the software application even if there are no failures or trigger events. By periodically polling, Applicant's system avoids the enormous data creation of the prior art processes (which trace all of the statistics generated by executing software applications), yet tracks the continuous operation of the software application and its presently executing functions.

Applicant's claimed system recites the following structure in claim 1:

> function identification means for automatically determining which function is presently executing in said identified software application;

In contrast to the monitoring system of the Harrison Patent, Applicant's claimed system bypasses the Reporting Application Interface and periodically polls the presently executing function within the identified software application to create a continuum of data to enable the user to view the continuing operation of the monitored system, as is recited in claim 1 as:

> polling means for bypassing said Reporting Application Interface and periodically polling said presently executing function in said identified software application to retrieve statistics data indicative of operation of said presently executing function in said identified software application;

As noted above, the monitoring system of the Harrison Patent fails to bypass the Reporting Application Interface and periodically polls "said presently executing function in said identified software application."

This fundamental difference between Applicant's system and that of the Harrison Patent is that Applicant's system tracks which functions are presently executing in the software application to create a history of operation of the monitored system, while the monitoring system of the Harrison Patent uses a standard Graphics User Interface of the Reporting Application Interface to generate standard reports via the GUI.

Therefore, Applicant believes that claim 1 is allowable under 35 USC §102(e) over the cited Harrison Patent, because the Harrison Patent fails to show or suggest the following structure recited in this claim:

> function identification means for automatically determining which function is presently executing in said identified software application;
>
> polling means for bypassing said Reporting Application Interface and periodically polling said presently executing function in said identified software application to retrieve statistics data indicative of operation of said presently executing function in said identified software application; and

Applicant also believes that independent claim 8 is allowable under 35 USC §102(e) over the cited Harrison Patent, since it is analogous to independent claim 1 in scope and content and is allowable for the reasons noted with respect to claim 1. In addition, Applicant believes that claims 2, 4 – 7, 9, and 11 – 14 are also allowable under 35 USC §102(e) over the cited Harrison Patent, because these claims depend on allowable base claims.

The Examiner also rejected claims 4 and 11 under 35 USC §103(a) as being unpatentable over the Harrison Patent in view of U.S. Patent No. 6,714,976 issued to Wilson (hereinafter "the

Wilson Patent"). Applicant believes that claims 4 and 11 are allowable under 35 USC §103(a) over the cited Harrison and Wilson Patents since these claims depend on allowable base claims.

In addition, the Wilson Patent describes the state of the art in monitoring and diagnosing systems in column 2, line 6 – column 3, line 6, noting that the prior art either continuously polls the monitored system and generates too much data or uses a custom API which overly filters the data. In response to these limitations in the prior art, the system of the Wilson Patent does not gather data on the present operation of the monitored system and only responds to a trigger event to gather data for monitoring and diagnostic purposes, then ceases to gather data. As noted in the Wilson Patent in column 3, lines 9 – 33:

> In accordance with principles of the invention is a method of monitoring a distributed computer system. Trigger events and associated data to be collected are defined. **The occurrence of one of the trigger events at a client is detected while monitoring a connection between a client and a first server. Client data is collected in accordance with the one trigger event at the client. A controller is notified of the detecting of the occurrence of the one trigger event.** The first server is notified of the occurrence of the trigger event. First server data is gathered by the first server, and the first server data is sent to the controller.
>
> In accordance with anther {*sic*} [another] aspect of the invention is a system for monitoring a distributed computer system. Machine executable code defines trigger events and associated data to be collected. Machine executable code detects occurrence of one of the trigger events at a client while monitoring a connection between a client and a first server. Machine executable code collects client data in accordance with the one trigger event at the client. Machine executable code notifies a controller of the detecting of the occurrence of the one trigger event. Machine executable code notifies the first server of the occurrence of the trigger event. Machine executable code gathers first server data by the first server, and machine executable code send s the first server data to the controller. [emphasis added]

Thus, the Wilson Patent is an error-reporting system, not a monitoring system since it is solely oriented to a trigger-based paradigm, where client data is gathered only once the trigger event is detected, and there is no data collection about the operation of the monitored system absent the trigger event.

In summary, Applicant believes that claims 1, 2, 4 – 9, and 11 – 14 are allowable over the cited references.

In view of the above remarks, Applicant believes the pending application is in condition for allowance. A one-month Petition For Extension Of Time and the appropriate fee are attached. If any additional fee is due, please charge our Deposit Account No. 50-1848, under Order No. 022058.0101PTUS from which the undersigned is authorized to draw.

Respectfully submitted,
PATTON BOGGS LLP

Dated: July 7, 2008

By: /James M. Graziano/
James M. Graziano
Registration No.: 28,300
(303) 830-1776
(303) 894-9239 (Fax)
Attorney for Applicant

Customer No. 24283